**Musical micro:bit**

**Lesson 2: Programming & debugging music**

**Introduction**

In this lesson, pupils develop their understanding of how to program the BBC micro:bit to play musical phrases by exploring and modifying programs that use the music blocks in MakeCode. They make use of their knowledge of repetition and inputs from previous units when writing programs to allow the micro:bit to play different musical phrases when certain conditions are met. They test and debug their programs before transferring to their device and exploring how to connect a micro:bit to speakers/headphones to play the music.

**Time:** @60 minutes

**Learning objectives**

* To use existing knowledge to improve programs
* To write and debug musical programs
* To experiment (tinker) with the micro:bit to make music

**Materials needed:** lesson presentation, laptops/computer with access to the MakeCode editor, algorithms from the previous lesson, printout of slide 11, starter, example & support hex files, micro:bits, battery packs, USB leads, crocodile clips, headphones/speakers.

**Lesson summary**

1. Introductory activity: represent that tune (10 mins)
2. Introducing musical programming (20 minutes)
3. From programs to algorithms (20 minutes)
4. Making the micro:bit musical (10 minutes)

**Introduction: Represent that tune (10 mins)**

* Use **slide 3** to invite pupils to recall the algorithms they wrote last lesson and discuss the questions with their partner before feeding back to the class,
* Run the [video clip](https://youtu.be/grKPI8fZ8yw) of a glockenspiel being played (**slide 4**) and ask pupils to write an algorithm that a person may have followed to play the musical phrase. Remind pupils to think about the feedback from their evaluations last session. Pupils can be supported by using printout of the octave on slide 11.
* Take a few examples and highlight pupils’ use of repetition. If repetition hasn’t been used, prompt thinking by asking questions such as, “*how else could we have communicated the musical phrase*?”

**Introducing musical programming (20 minutes)**

* Display **slide 5** and invite pupils to use their current understanding of the micro:bit and MakeCode to make statements about the program (see speaker notes for suggestions).
* Use the [link](https://makecode.microbit.org/#pub:_12E3wp3JoYoP) on the slide (the image of the **starter hex file** program) to open up the same program using the MakeCode editor and invite suggestions for how to get the music to be played.
* Explain to pupils they are going to write a program with the same output using fewer blocks. Allow pupils to think/pair/share their initial responses before letting them work with a partner to rewrite the program. The program should make use of repetition (see slide 12 for example).
* Once pupils have had the opportunity to rewrite the program, ask them to share their programs and comment on how they were able to create the same output using fewer blocks. Identify the concept that pupils used (repetition) and the benefits of using it: fewer instructions are used so less time is spent writing and debugging the program (**slide 6**) and it is more efficient for the micro:bit to run.

.

**From programs to algorithms (20 minutes)**

* Explain to pupils that they are going to write a program to play the musical phrases they composed in the previous lesson. Discuss the need for each phrase to start when a different condition is met (**slide 7**).
* Pupils work with a partner to program the micro:bit to play their musical phrases composed in the previous lesson. Each musical phrase should start with a different input. Remind pupils to test their program frequently and debug when necessary (see **example hex file** with lesson downloads and on slide 12).

**Making micro:bit musical (10 minutes)**

* *This activity requires access to physical micro:bits and associated hardware, if you do not have the required resources, skip this section.*
* Give out headphones/speakers, micro:bits, battery packs, crocodile clips and USB connectors and display **slide 8**. Invite a pupil to model how to download and transfer the program to a micro:bit.
* Explain that pupils are going to experiment (tinker) to find out how to get their musical phrases to be played when the micro:bit is not connected to the computer. Display **slide 9** and explain that this requires a lot of the information they need and give pupils time to experiment.
* Invite pupils to share their findings and explain how they made use of the image on **slide 9**. Ask pupils how they used trial and error to solve the problem, highlighting the importance of ‘failure’ to learning.
* If you wish, used **slide 10** to review the learning objectives of the session.

**Extension ideas**

* Pupils could play a duet with micro:bits. The pupils play their musical phrase on the glockenspiel alongside the micro:bit playing it through speakers. This could be recorded and shared.

**Differentiation**

**Support:** Pupils could use a partially completed program (**see support hex** file) or complete the task as a guided group with adult support.

**Stretch & challenge:** Pupils could use the ‘add comment’ function in the MakeCode editor (right-click) to annotate their program to explain their use of repetition and selection in the program.

**Opportunities for assessment**

* Informal observation of pupils understanding of repetition, programming and tinkering through class and group discussions and activities.
* More formal assessment of pupils’ algorithms and programs if wished.